Telecommun Syst (2007) 35: 67–86
DOI 10.1007/s11235-007-9041-5

A trajectory-based selective broadcast query protocol for large-scale, high-density wireless sensor networks

Christopher R. Mann · Rusty O. Baldwin · Jeffrey P. Kharoufeh · Barry E. Mullins

Published online: 12 September 2007

Abstract We present a small-footprint search protocol designed to facilitate any-type queries for data content and services in large population, high-density wireless sensor networks. Our protocol, termed Trajectory-based Selective Broadcast Query (TSBQ), works in conjunction with time division multiple access- or schedule-based medium access control protocols to reduce per-query energy expenditure. We compare the performance of TSBQ to unicast- and local broadcast-based search algorithms and also determine a critical node density based on the energy expended by nodes to transmit and receive. Minimal energy expenditure is achieved by determining the optimal number of data/service replicates and the number of nodes designated to receive each query transmission. Numerical results indicate that TSBQ significantly reduces the total energy expenditure of a network as compared to unicast and local broadcast-based search protocols.

The views expressed in this paper are those of the authors and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U.S. government.

C.R. Mann (✉) · R.O. Baldwin · B.E. Mullins
Department of Electrical and Computer Engineering, Air Force Institute of Technology, 45433-7765 Wright-Patterson Air Force Base, USA
e-mail: christopher.mann@afit.edu

R.O. Baldwin
e-mail: rusty.baldwin@afit.edu

B.E. Mullins
e-mail: barry.mullins@afit.edu

J.P. Kharoufeh
Department of Mechanical and Industrial Engineering, 334 Snell Engineering Center, Notheastern University,
360 Huntington Avenue, Boston, MA 02115, USA
e-mail: jkharoufeh@coe.neu.edu

Keywords Search algorithms · Service and content location · Wireless sensor networks

1 Introduction

Wireless sensor networks (WSN) are formed through the cooperation of hundreds or thousands of small sensing devices. These devices, called *nodes*, are linked via a wireless transmission medium to perform tasks in a distributed manner. Nodes typically have limited energy stores, transmission range, local storage capacity, and computational ability.

It is envisioned that wireless sensor networks will soon be composed of hundreds of thousands or even millions of nodes. As these computing devices become smaller in size, they will be embedded in raw materials to construct buildings, bridges, houses, and roads. They will perform functions such as monitoring the integrity and security of structures and territory as well as tracking traffic patterns, accidents, and weather. In an emergency situation, such networks will alert the appropriate personnel autonomously, reducing response time and saving lives and assets. As sensor networks become ubiquitous, disparate networks will eventually merge due to proximity and necessity. In time, a network monitoring the structural integrity of a building might, for example, communicate with the network monitoring nearby roads as well as the networks embedded in other buildings.

As the size and scale of wireless sensor networks continue to grow, two characteristics will be critical to maintaining their viability. First, high node densities (i.e., those for which each node has a large number of one-hop neighbors) will be necessary to meet an increasing demand for high-precision sensor data while simultaneously providing redundant communication paths throughout the network.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **SEP 2007**	2. REPORT TYPE	3. DATES COVERED **00-00-2007 to 00-00-2007**

4. TITLE AND SUBTITLE
A trajectory-based selective broadcast query protocol for large-scale, high-density wireless sensor networks

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology ,Department of Electrical and Computer Engineering,2950 Hobson Way,Wright Patterson AFB,OH,45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT
We present a small-footprint search protocol designed to facilitate any-type queries for data content and services in large population, high-density wireless sensor networks. Our protocol, termed Trajectory-based Selective Broadcast Query (TSBQ), works in conjunction with time division multiple access- or schedule-based medium access control protocols to reduce per-query energy expenditure. We compare the performance of TSBQ to unicastand local broadcast-based search algorithms and also determine a critical node density based on the energy expended by nodes to transmit and receive. Minimal energy expenditure is achieved by determining the optimal number of data/service replicates and the number of nodes designated to receive each query transmission. Numerical results indicate that TSBQ significantly reduces the total energy expenditure of a network as compared to unicast and local broadcast-based search protocols.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**20**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

High node density also results in increased average lifetime per unit density of the network, a favorable property in networks composed of large numbers of low-cost, unreliable nodes [32]. Second, small-footprint, scalable, energy-efficient applications will remain a critical enabling technology. Key among these critical applications is the capability of nodes to locate data and services within the network. Due to the distributed nature of data collection in WSNs, no single node is likely to have all the information necessary to complete a particular task. However, locating information requires nodes to expend precious energy reserves thereby reducing node and network lifetime. Developing an energy-efficient means for nodes to both advertise the availability of data or services and to locate these items within the network is the focus of this paper.

The main contributions of this research may be summarized as follows. First, we develop a small-footprint search protocol, Trajectory-based Selective-Broadcast Query (TSBQ), which significantly reduces the total energy used to advertise and locate data and services within a network as compared to unicast and local-broadcast search methods. Second, we provide an analytic model for the expected total energy expended by TSBQ and show how to minimize the expected total energy expended by determining simultaneously the optimal number of agent replicas and the number of nodes that should be designated as receivers for each query transmission. Using our model, we predict the performance variance of rumor routing-based search protocols and offer a means to minimize this variance. Third, by means of a simulation model, we evaluate the performance of TSBQ and consequently, we propose further refinements to the protocol. Finally, we elucidate the effect of network boundaries and incorporate these effects into the mathematical model.

The remainder of this paper is organized as follows. In Sect. 2, we discuss related work. In Sect. 3, we develop and analyze a mathematical model for the expected total energy expenditure of the TSBQ protocol. The results of simulation experiments with large, high-density networks are presented in Sect. 4. Based on the results of these experiments, we propose improvements to the protocol and mathematical model. Section 5 provides conclusions and directions for future work.

2 Related work

The original rumor routing protocol [4] and several of its variants [1, 2, 6, 27] are most closely related to our proposed Trajectory-based Selective Broadcast Query (TSBQ) search protocol. In rumor routing [4], both queries and resource advertisements (called "agents") travel from node to node by means of a modified random walk. Nodes in receipt of an agent are capable of answering related queries from other nodes. In [6], rumor routing's dual problems of spiraling agent/query routes and ever-increasing packet size (due to the need to record previously-visited nodes to prevent backward paths) are solved by forwarding agents and queries using straight-line routing (SLR). REDMAN [2] is similar to SLR in that agents and queries are forwarded along straight-line trajectories. However, resource replicas are stored only at every kth node along the agent's path; the remaining intermediate nodes store a pointer to the nearest available replica. Zonal Rumor Routing [1] is an extension of rumor routing that partitions the network into artificial zones for purposes of choosing intermediate nodes for agent/query routing. Neighboring nodes assigned to unvisited zones are favored when choosing an agent or query's next hop, thus improving the probability of a successful query. With respect to our research, however, we have noted that there are currently no analytic models of rumor routing-based search protocols that permit determination of optimum resource replication levels based on node hardware characteristics and resource popularity. Moreover, none of these protocols take advantage of the power of broadcast transmissions, nor do they incorporate a feedback-driven caching mechanism to improve latency and decrease the energy expended by subsequent queries.

Quorum-based search protocols (see, for example, [12, 16, 25]) also seek to facilitate intersection between queries and their corresponding agent trajectories. This is typically accomplished by forwarding along straight-line paths in each of the four cardinal directions. For example, in GCLP [27] both agents (called "content advertisements") and queries are propagated along straight-line trajectories in the north–south and east–west directions. This method guarantees intersection of a query with at least one Content Location Server (i.e., a node aware of the location of a specific resource). Quorum-based schemes can also achieve a measure of energy efficiency by aggregating advertisements at each node prior to transmission. However, most quorum-based schemes require nodes to maintain sizeable stores of information regarding the location of distant nodes; in mobile networks, this information must be frequently updated or the node risks returning stale information in response to a query. Also, quorum-based search protocols treat resources with equivalent importance. Both popular and unpopular items consume the same amount of network storage capacity, and the mean energy and latency required to locate both popular and unpopular items are the same. Our research indicates this paradigm forces over-representation of unpopular items within the network's aggregate storage capacity and increases the total energy expended for popular item queries.

Rumor routing and quorum-based approaches can be described as "geocentric" because the dispersal of resource advertisements and/or replicates is based on network topology or direction. Such approaches differ from "data-centric"

search algorithms in that the requesting node has no knowledge of the location of the desired resource when it issues the query. As an alternative, resources in data-centric networks are self-organized to facilitate answering queries. Data-centric methods, including [20, 21, 23], remove uncertainty regarding the location of resources by forwarding data to a pre-defined network location, node, or group of nodes based on specific data characteristics. Queries are forwarded directly to the node or nodes responsible for caching related data, thus decreasing latency and energy expenditure. However, data-centric algorithms are not without their challenges. Most importantly, all nodes require access to a common hash table to determine where to send their resource advertisements and queries. "Hotspotting," rapid energy depletion, storage capacity overload, and/or congestion can occur if a single node or location becomes the repository for a large amount of network data or queries, and network-wide updates to the hash table—when needed—are costly in terms of energy expenditure. Scalability can also be problematic unless a load balancing method, such as a distributed hash table, is used (see, for example, [14, 19, 22, 24, 33]). Finally, because nodes are subject to random failure, data redundancy is a favorable characteristic to prevent data loss. However, data-centric algorithms tend to group related data in close proximity, thus increasing the severity of data loss in the event of node failure or network partitioning. To achieve redundancy in data-centric networks, data may be replicated at nodes located in close proximity to the hashed location [23] or dispersed throughout the network in a geocentric-type approach. Although data-centric approaches can be effective, we favor the simplicity, flexibility, redundancy, and data dispersion achieved by geocentric approaches in networks composed of large numbers of unreliable nodes.

There also exist ongoing and relevant efforts to develop efficient replication and search strategies in unstructured peer-to-peer networks [3, 7–9, 15]. However, these efforts are primarily focused on reducing query latency instead of increasing energy efficiency because the computers in those types of networks are less constrained by available energy, local storage, and computational capability. The means by which to evaluate tradeoffs between important network parameters—including the number of agent replicas stored in the network, total network storage capacity, hardware power requirements, and node density—has received little attention in the open literature. This research closes that gap by providing a means to evaluate the effects of these parameters on overall energy savings, effective total network storage capacity, query response variance, and query latency.

Although TSBQ is inspired by traditional rumor routing, the following characteristics make it unique:

- TSBQ is the only WSN search protocol to minimize the total expected energy expenditure of the network by analytically determining the optimum number of resource replicates created by each agent. Additionally, TSBQ leverages the broadcast nature of wireless transmissions to query multiple nodes per transmission, thereby reducing total energy expenditure.
- TSBQ specifically accounts for resource popularity as well as the energy expended by nodes both to listen and to receive when determining the appropriate number of receivers and the number of nodes informed via agents. Additionally, TSBQ accounts not only for the energy expended to inform the network via an agent and locate the desired information via a query but also for the energy expended to return the response to the originating node. Achieving maximum energy savings requires optimizing each of these sources of energy expenditure simultaneously.
- Nodes need only maintain one-hop neighbor information to eliminate redundant node querying. Although a node may receive a *reissued* query more than once (cf., Sect. 4), this can be eliminated by permitting nodes to ignore the PQN's broadcast during the next transmission period if they received the query from the QN in a previous transmission slot.
- TSBQ reduces network congestion by limiting retransmissions of the query packet to a single query node (QN), thus avoiding the inherent difficulties and inefficiencies associated with network flooding.
- TSBQ includes a feedback-driven caching mechanism to reduce search latency for popular data/services. This mechanism requires negligible additional energy expenditure by the network.

3 Protocol description

It is well known that nodes can conserve energy resources by turning off transmitting and receiving hardware when not in use [13, 18, 21, 28]. Several medium access control (MAC) protocols such as S-MAC [31], D-MAC [13], T-MAC [28], and TRAMA [18] achieve energy savings in this manner. TSBQ takes advantage of node hardware characteristics and the energy savings of time division multiple access (TDMA)-based MAC protocols to determine the appropriate advertising and query strategy for the network. Although all nodes must participate in the MAC's contention period to coordinate transmission and reception schedules, nodes not designated to transmit or receive during a given transmission period are permitted to enter a low-power sleep mode. We then seek to minimize the total energy expended by simultaneously determining the appropriate number of receivers designated by the MAC during each transmission period and the optimum proportion of resource replicates.

3.1 TSBQ overview

When discussing the means to propagate and locate information within a network, we adopt and expand much of the terminology of Braginsky and Estrin [4]. *Agents* are packets transmitted by *witness nodes* to advertise the availability of specific services or data. *Informed nodes* have received an agent transmission and stored the agent's content in a local *event table*. A node seeking data or a particular service is the *origin query node* (OQN), and nodes that relay query packets on behalf of the OQN are *query nodes* (QN). OQNs and QNs transmit *queries*, packets that "roam" the network in search of specific services or data. *Receiving nodes* (RN) adjust their sleep cycles to accommodate the transmission schedules of neighboring OQN/QNs when designated by the OQN/QN to receive a query transmission. When a query is received by an informed node, the node generates a *response* that is returned to the OQN. The response may contain the specific data requested by the end-user or simply provide the location of the desired data or service.

Two basic principles motivate the development of TSBQ. First, we seek a balance between the energy expended to inform the network of an event or service via an agent and the energy required to locate an informed node via a query. If too few nodes are informed, less energy is used to transmit agents and the network storage burden is decreased. However, a query will likely expend additional energy to locate an informed node thereby negating any potential energy savings. Conversely, if too many nodes are informed, the amount of energy expended for each query is reduced, but the energy required to propagate each agent is increased and a larger portion of the network's aggregate storage capacity is consumed. Second, when querying neighboring nodes, we must balance the number of nodes that receive each query transmission and the energy expended by these nodes to receive the query. If too few nodes receive the query, additional transmissions may be required to locate an informed node. By contrast, if too many nodes receive the query, an informed node may be located with lower latency, but the uninformed receiving nodes still pay a cost for receiving the query packet.

The TSBQ search protocol consists of the following steps:

1. A node witnesses an event and generates an agent to inform an additional $(\alpha N - 1)$ nodes, where N is the number of nodes in the network. To ensure the value $(\alpha N - 1)$ is integral, $\alpha \in \{1/N, 2/N, \ldots, (N-1)/N\}$.
2. An OQN generates a query and chooses a random direction (trajectory) for routing. Based on this trajectory, the OQN chooses the next *potential query node* (PQN) from among its one-hop neighbors using the Most Forward within Range (MFR) criterion (Fig. 1) [26].

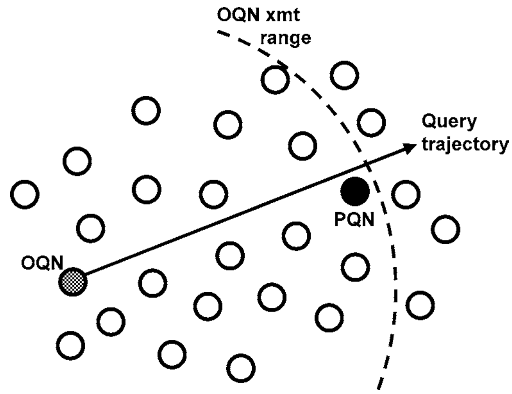


Fig. 1 The OQN chooses the PQN using MFR

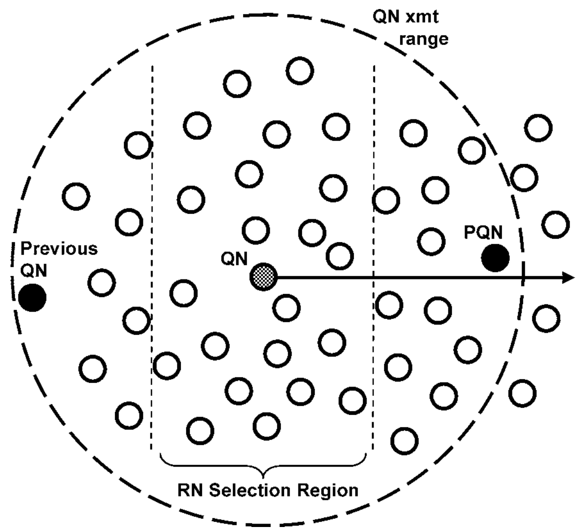


Fig. 2 RN selection region (simplified transmission model)

3. The OQN/QN randomly selects $(\delta' - 1)$ RNs from among its neighbors that are closer to itself than the PQN (Fig. 2), where δ' is a positive integer no greater than the cardinality of the node's neighbor set, δ. (The means to determine the optimum value of δ' is discussed in Sect. 3.)
4. Transmission/reception coordination between the OQN/QN and RNs is achieved via a TDMA- or schedule-based MAC protocol during the contention period. The OQN/QN sets the transmission-reception schedule for its neighbors and designates the RNs. Nodes not designated as a QN, PQN, or RN enter sleep mode to conserve energy during the appropriate transmission period(s).
5. The OQN/QN broadcasts the query to the PQN and the designated RNs (a total of δ' receivers per query transmission).
6. If no response is received from the PQN or RNs (i.e., the query fails to locate an informed node), then the PQN becomes the next QN. The new QN chooses a PQN using MFR along the designated trajectory. The process returns

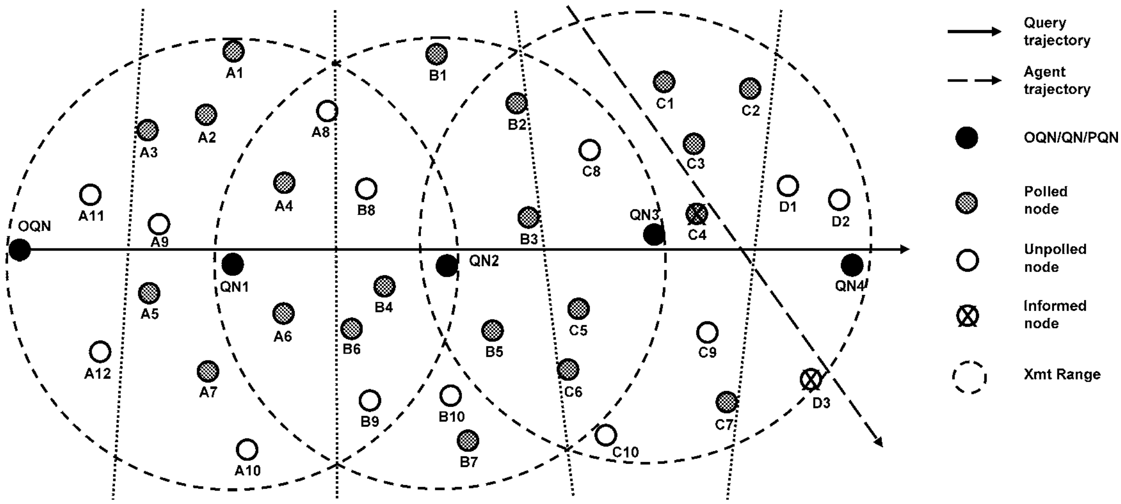


Fig. 3 Graphical depiction of the TSBQ protocol

to Step 3 and repeats until the query is successful or terminated.

7. If at least one PQN or RN is informed, the node transmits the desired information to the QN. The response is then returned to the OQN via MFR routing along the trajectory defined by the positions of the QN and OQN. The query is terminated by the PQN once it overhears the response transmitted by the QN.
8. A feedback-driven caching mechanism may be incorporated to enable intermediate nodes along the route from the informed node to the OQN to add the information in the response to their own event tables. This mechanism is discussed in Sect. 4.

The partial network diagram in Fig. 3 is a graphical depiction of the TSBQ protocol. The black arrow is the OQN's randomly-chosen query trajectory, the black circles are the sequence of PQN/QN nodes responsible for transmitting the query at each hop, and the gray circles designate the RNs randomly polled by a QN to determine if they have a corresponding agent. The dashed arrow represents the trajectory of the desired agent, and an "X" indicates a node is informed. Nodes C4 and D3 in Fig. 3 have received and stored a copy of the agent sought by the OQN. In this example, each node has approximately $\delta = 18$ one-hop neighbors, and $\delta' = 8$. The means to analytically determine δ' is discussed in Sect. 3.3.

When a node needs a non-local resource yet has no knowledge of the resource's location, the node designates itself as the OQN and randomly picks a query trajectory. Based on this query trajectory, the OQN selects the PQN (node QN1 in Fig. 3) and randomly chooses $(\delta' - 1) = 7$ neighbors (i.e., RNs) from among those nodes closer to itself than the PQN. After coordinating with its neighbors during the MAC contention period, the OQN transmits the query to the PQN and the RNs. The OQN's remaining neighbor nodes are permitted to sleep during this transmission period. If neither the PQN nor the seven RNs polled by the OQN can answer the query, the PQN will query a subset of its neighbors on behalf of the OQN. Although not depicted in Fig. 3, the OQN's query is unsuccessful; therefore, node QN1 must forward the query.

Based on the query trajectory chosen by the OQN, node QN1 identifies node QN2 as the PQN and randomly selects nodes A1–A7 as RNs. Since neither QN2 nor A1–A7 are informed, QN1's query fails, and QN2 assumes responsibility for the next query transmission. QN2 chooses a PQN (QN3) based on the specified query trajectory and selects seven RNs (B1–B7). Since none of these nodes hold a copy of the desired agent, QN2's query also fails.

Once QN3 recognizes QN2's query has failed, it identifies the PQN (QN4) and chooses seven RNs (C1–C7). Upon polling these nodes, node C4 responds with the desired information. QN3 uses this information to generate a response, determines the appropriate response trajectory, and returns the response to the OQN. When QN4 overhears the response transmitted by QN3, it terminates the query.

During each query transmission, it is possible that an informed node is a neighbor of the QN but not located because the node was not chosen as a PQN or RN. If this occurs, the result is increased delay in returning a response to the OQN and additional transmissions. Eliminating this possibility can only be achieved by transmitting the query to *all* neighboring nodes. However, in Sect. 3.4, we show the total expected energy expended by the network to answer a query is minimized by choosing a subset of a node's neighbors as receivers when the node density exceeds a specific threshold.

3.2 Analytic model of TSBQ energy expenditure

Three primary sources of network energy expenditure are required to generate a successful response to a query: agent transmission/reception, query transmission/reception, and response transmission/reception. Achieving the minimum energy expenditure per successful query requires balancing these elements. We discuss each individually in the following subsections.

Although the MAC protocol is also a source of energy expenditure, evaluation of these protocols within the context of our model is beyond the scope of this paper; we assume the energy expended by the MAC protocol per successful transmission is approximately constant for networks of a given size and density. Our primary performance metric—the sum total energy required for agent transmission/reception, query transmission/reception, and response transmission/reception—measures the total amount of energy expended by nodes during contention-free transmission periods.

3.2.1 Agent transmission/reception

Traditional rumor routing assumes each node within range of an agent transmission receives the agent and adds the event to its local event table. This results in a "thick line" of informed nodes in the network [4]. However, in high-density networks, this approach has two disadvantages: first, a large percentage of the total network storage capacity is consumed by these agents. Event tables of nodes located near active areas of the network will likely reach their capacities quickly, requiring a replacement strategy for event table entries—an undesirable alternative. Second, unless the agent time-to-live (TTL) value is high, an agent may not be transmitted to distant regions of the network. This means large portions of the network have no informed nodes (i.e., a low spatial dispersion of informed nodes). As a consequence, networks using traditional rumor routing techniques may be unable to locate an informed node without large energy expenditure.

To increase the spatial dispersion of informed nodes while simultaneously minimizing the number of transmissions, we propose that agents be forwarded along straight-line trajectories in a manner similar to the techniques used in [2, 17, 27]. Additionally, to minimize local storage requirements, each agent transmission is intended for exactly one receiving node (i.e., agents are transmitted in a unicast fashion). Coordination between transmitting and receiving nodes is achieved via a TDMA- or schedule-based medium access control protocol, such as T-MAC, during the MAC protocol's contention period. During the transmission period, all nodes within range of the agent transmission not designated as receivers deactivate their receiving hardware to conserve energy. The intended receiving node is chosen using MFR to eliminate routing loops [26].

In the event a node cannot forward an agent along the desired trajectory (e.g., due to encountering a network boundary), the node will randomly choose a new forwarding trajectory for the agent. (Alternatively, if the agent cannot be forwarded due to a void or obstacle within the network, a face routing scheme—such as Greedy Perimeter Stateless Routing [11]—can be used to circumvent this region until the desired trajectory can be resumed. However, in the design space of large-scale, high-density networks using MFR, the probability of encountering a void is small [30]. Therefore, this occurrence is not included in the development of the mathematical model.) Each agent is forwarded to exactly $(\alpha N - 1)$ unique nodes, thus ensuring there are αN informed nodes.

Once a node receives an agent, the node makes an entry in its event table that includes the type of service/data advertised, the location of the witness node, and a copy of the data (if available). Although any node that overhears an agent transmission may add the agent to its event table, we advocate the unicast transmission of agents between nodes and the use of MFR to select receivers as a means to promote the maximum physical distance between identical event table entries. This reduces the probability that large numbers of informed nodes are found only within limited portions of the network.

If we let A denote the total energy used to propagate each agent, then for large networks such that $\alpha \ll 1$, the expected total energy used to propagate each agent is

$$E[A] = (E_{xmt} + E_{rcv}) \cdot (\alpha N - 1), \tag{3.1}$$

where E_{xmt} is the energy expended by a node to transmit a packet, and E_{rcv} is the energy expended to receive a packet.

3.2.2 Query transmission/reception

When a node needs access to services or data but has no corresponding entry in its event table, the node generates a query. Because nodes may selectively activate and deactivate their receiving hardware, the node's query transmission may be received by one, some, or all of its one-hop neighbors simultaneously. If we momentarily assume that informed nodes are spatially uniformly distributed throughout the network and disregard the effect of network boundaries (assumptions we will revisit in Sect. 4), the number of informed nodes that are also neighbors of each QN is a binomial random variable.

Let Y be the number of informed nodes within one-hop distance of the QN. If a QN has δ neighbors and a corresponding query is transmitted to δ' of these neighbors, $0 < \delta' \le \delta$, then the probability of failing to find an informed

node is obtained by

$$\Pr\{Y=0\} = \binom{\delta'}{0}\alpha^0\left(1-\frac{\alpha N}{N-1}\right)^{\delta'} = \left(1-\frac{\alpha N}{N-1}\right)^{\delta'}, \tag{3.2}$$

and the probability of finding at least one informed node is

$$\Pr\{Y>0\} = 1-\left(1-\frac{\alpha N}{N-1}\right)^{\delta'}. \tag{3.3}$$

Note that we assume a node does not generate a query for a particular service or data if it is already informed. As a consequence, the probability that an uninformed node's neighbor possesses the data of interest is slightly greater than α.

In TSBQ, queries are forwarded along straight-line trajectories in a manner similar to that used for agents. However, in contrast to agent transmissions, queries are broadcast to a subset of each node's neighbors. Nodes that have not been designated to receive a particular query transmission turn off their receivers to conserve energy. The use of straight-line routing trajectories increases the probability that a subset of the QN's neighbors have not yet received the current query compared to random walk methods. Therefore, the probability of finding an informed node increases with each hop of the query along its assigned trajectory. Let Z_j be a Bernoulli random variable denoting success or failure of the jth query hop (transmission) such that $Z_j=0$ when the jth query hop fails to locate an informed node and $Z_j=1$ otherwise. If a query is broadcast to a unique set of δ' receivers at each hop in its path, the probability that the jth query transmission fails to locate an informed node is

$$\Pr\{Z_j=0\} = \left(1-\frac{\alpha N}{N-1-(j-1)\delta'}\right)^{\delta'}, \quad j\ge 1. \tag{3.4}$$

If an informed node is found on the jth hop, then an informed node was not located on the previous $(j-1)$ hops because a query is not propagated further once an informed node is found. (Recall that we are performing an *any*-type search; therefore, the search is concluded when at least one copy of the desired information is located.) Consequently, the probability of locating an informed node for the first time on the jth hop is

$$\Pr\{Z_1=Z_2=\cdots=Z_{j-1}=0, Z_j=1\}$$
$$= \begin{cases} 1-\left(1-\frac{\alpha N}{N-1}\right)^{\delta'}, & j=1, \\ \left[1-\left(1-\frac{\alpha N}{N-1-(j-1)\delta'}\right)^{\delta'}\right] & \\ \quad\times\prod_{i=1}^{j-1}\left(1-\frac{\alpha N}{N-1-(i-1)\delta'}\right)^{\delta'}, & j\ge 2. \end{cases} \tag{3.5}$$

Clearly, sensor networks are comprised of a finite number of nodes. Assuming a query can be propagated without encountering a network boundary, the maximum number of query transmissions that can be made to unique neighboring nodes before locating *at least one* informed node is

$$k := \left\lfloor \frac{N(1-\alpha)-1}{\delta'} \right\rfloor + 1, \quad \alpha\in\{1/N, 2/N, \ldots, (N-1)/N\}. \tag{3.6}$$

Equation (3.6) assumes that at least one node in the network has not received a copy of the agent; otherwise, there would be no need for a node to generate a query. Let $X_{\alpha,\delta'}$ denote the random number of transmissions required to find an informed node for fixed values of α and δ'. Then the probability of requiring j query transmissions is

$$\Pr\{X_{\alpha,\delta'}=j\}$$
$$= \begin{cases} 1-\left(1-\frac{\alpha N}{N-1}\right)^{\delta'}, & j=1, \\ \left(1-\left(\max\left\{1-\frac{\alpha N}{N-1-(j-1)\delta'}, 0\right\}\right)^{\delta'}\right) & \\ \quad\times\prod_{i=1}^{j-1}\left(1-\frac{\alpha N}{N-1-(i-1)\delta'}\right)^{\delta'}, & 2\le j\le k \end{cases} \tag{3.7}$$

and the expected value of $X_{\alpha,\delta'}$ is

$$E[X_{\alpha,\delta'}] = \sum_{j=1}^{k} j\cdot\Pr\{X_{\alpha\delta'}=j\}. \tag{3.8}$$

Let Q be the energy expended by the network to locate an informed node. The use of straight-line trajectories for forwarding queries—and assuming no redundant polling of nodes—implies that the expected energy used to forward a query can be derived from (3.7) as

$$E[Q] = n\cdot(E_{xmt}+\delta'\cdot E_{rcv})\cdot E[X_{\alpha,\delta'}], \tag{3.9}$$

where n is the total number of unique queries generated by n OQNs to locate a particular agent. Note that the number of informed nodes, αN, is assumed to be constant for all n queries. Although the number of informed nodes should increase as queries are answered, we make no temporal assumptions regarding the generation of queries or responses. Hence, (3.9) represents an upper bound for the expected energy expended by the network to locate an informed node. Additionally, the value of n may be set prior to deployment based on analysis of the network's application(s), or it may be updated dynamically if, for example, one or more nodes recognize the number of unique requests for a particular resource exceeds a specified threshold. Alternatively, a feedback-driven caching mechanism can be used (cf., Sect. 4.3).

3.2.3 Response transmission/reception

Once the desired information is located, the response is returned to the OQN. Although we assume intermediate nodes in the response path are chosen using MFR along the straight-line trajectory defined by the current QN and OQN, there are several energy-efficient routing protocols that could perform this function. Most notably, Span [5] and GAF [29] provide point-to-point routing services and are specifically designed to reduce energy expenditure by maximizing the number of nodes in the sleep state.

Let R denote the energy used by the network to return a response to the OQN. Assuming the query does not encounter a network boundary prior to locating an informed node, the expected number of transmissions to return the response is identical to the expected number of query transmissions required to locate the informed node. Then the expected energy required to return n responses to n OQNs is

$$E[R] = n \cdot (E_{xmt} + E_{rcv}) \cdot E[X_{\alpha,\delta'}]. \tag{3.10}$$

3.2.4 Expected energy requirement

The total energy T required to propagate an agent, its associated query(ies), and response(s) is the sum of (3.1), (3.9), and (3.10). An additional transmission and reception must be added for each query since an informed node, once located, must advise the current QN the desired information has been found. Therefore, the expected total energy expended by the network to generate n unique responses is

$$\begin{aligned} E[T] &= (\alpha N - 1 + n)(E_{xmt} + E_{rcv}) \\ &\quad + (2nE_{xmt} + n(\delta' + 1)E_{rcv}) \cdot E[X_{\alpha,\delta'}]. \end{aligned} \tag{3.11}$$

3.3 Minimizing expected total energy expended

The main objective of the TSBQ protocol is to minimize the expected total energy expended by the network to generate n successful responses to n queries for the desired data/service. Therefore, whenever E_{rcv}, E_{xmt}, N, and n are known, we seek to select the optimal pair (α, δ') that minimizes (3.11).

We now formalize the problem and its solution procedure. To emphasize the explicit dependence of (3.11) on the decision variables α and δ', let $f(\alpha, \delta') \equiv E[T]$ denote the expected total energy expended by the network. The mathematical programming formulation is as follows:

Minimize $f(\alpha, \delta')$

Subject to:

$$\alpha \in \{1/N, 2/N, \ldots, (N-1)/N\},$$

$$\delta' \in \{1, 2, \ldots, \delta\}.$$

For a finite network, $f(\alpha, \delta')$ is a discrete function on a feasible region with $(N-1) \cdot \delta$ possible solutions. Therefore, the mathematical program is a straightforward discrete optimization problem in which the minimum energy expenditure may be obtained by enumerating all possible combinations of (α, δ'), and then choosing the (α, δ') pair that results in the least total energy expended. We refer to the pair of α and δ' values that result in the minimum expected energy expenditure as (α^*, δ'^*). A partial graph of the objective function for a 5000-node network is shown in Fig. 4. Note that we normalize the expected total energy expended by the energy expended for node transmission; we also assume $0 < E_{rcv} \le E_{xmt}$. The E_{rcv}/E_{xmt} ratio is defined by the hardware characteristics of the nodes and sizes of the transmitted packets. It can also include the energy expended by the MAC layer for transmissions and retransmissions.

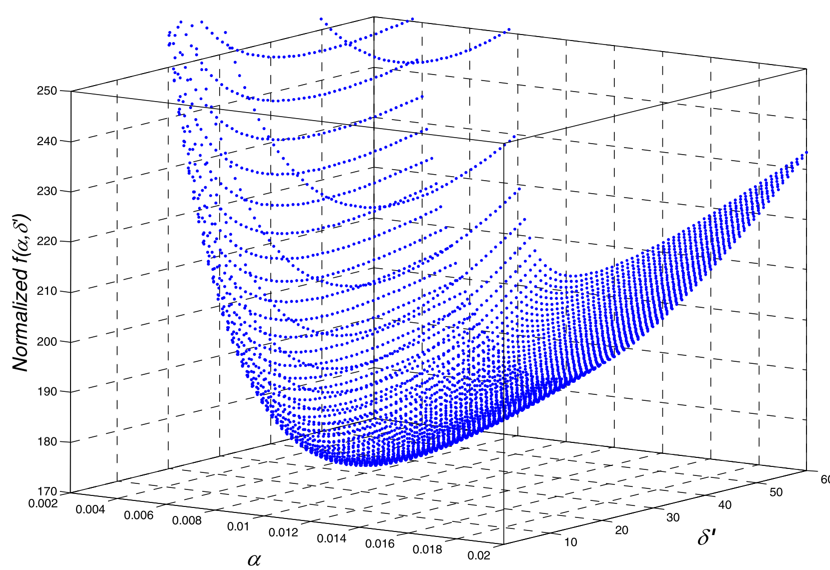


Fig. 4 Plot of $f(\alpha, \delta')$, $N = 5000$, $n = 1$, $E_{rcv}/E_{xmt} = 0.7$

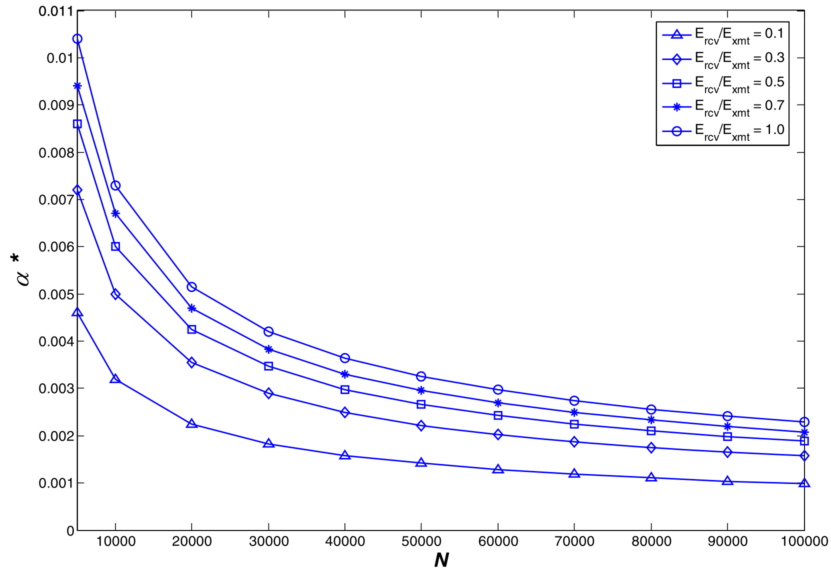


Fig. 5 Effect of N and E_{rcv}/E_{xmt} on α^*, $n = 1$

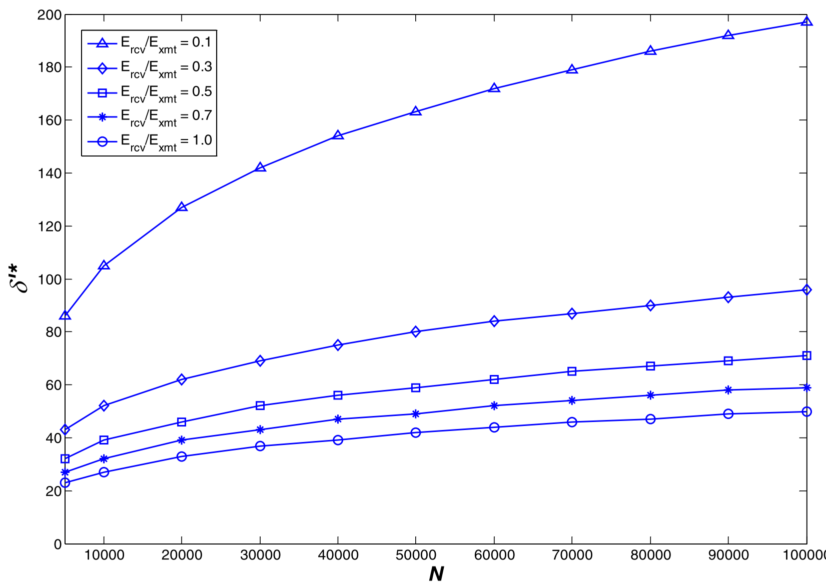


Fig. 6 Effect of N and E_{rcv}/E_{xmt} on δ'^*, $n = 1$

We now examine the effect of increased network size and various E_{rcv}/E_{xmt} ratios on the optimal (α, δ') pair. The results of this analysis for a wide range of network sizes are shown in Figs. 5 and 6 for the single-query case (i.e., $n = 1$), and the minimum expected total energy expended is shown in Fig. 7. For example, if $E_{rcv}/E_{xmt} = 0.5$, (α^*, δ'^*) for a 50000-node network is (0.00266,59), and the expected total energy expended (normalized) is 419.6.

3.4 Approximating the optimal solution

Although (α^*, δ'^*) can be obtained for a network of fixed size, density, and E_{rcv}/E_{xmt} ratio via explicit enumerations, this method imposes a high computational requirement for N very large. In the worst case, the optimization program requires $O(N)$ floating-point additions, $O(N^2)$ floating-point multiplications, and $O(N^2)$ floating-point exponential operations. For extremely large, dense, networks, it may not be feasible to carry out this analysis. Additionally, the parameters that characterize a newly deployed network will almost certainly change during the network's useful lifetime, requiring the optimal solution to be periodically updated. Ideally, it would be advantageous to express α^* and δ'^* as functions of N and E_{rcv}/E_{xmt}.

Regression analysis of the curves in Figs. 5 and 6 reveals that the power model provides an excellent fit to our numerical results, yielding correlation coefficients greater than 0.999. The generalized power model is

$$A = B \cdot C(x)^p, \tag{3.12}$$

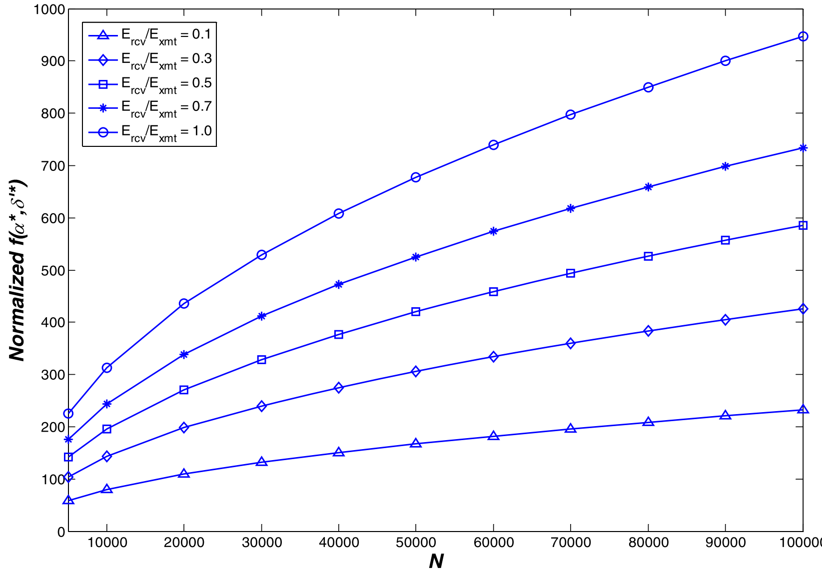


Fig. 7 Expected minimum energy expended using (α^*, δ'^*), $n = 1$

where A is the dependent variable, $C(x)$ is the independent variable, and B and p are constants. The following equations are used to determine α^* and δ'^* as a function of the network size N:

$$\alpha^* = b_1 \cdot N^{p_1},\\ \delta'^* = b_2 \cdot N^{p_2}, \tag{3.13}$$

where b_1, b_2, p_1, p_2 are constants for a fixed E_{rcv}/E_{xmt} ratio.

The regression analysis reveals the following: first, the value of α resulting in the smallest total energy expenditure for a fixed E_{rcv}/E_{xmt} ratio is inversely proportional to the square root of N (i.e., $p_1 \approx -0.5$), and b_1 increases as the E_{rcv}/E_{xmt} ratio increases. Hence, as network size increases, the minimum expected energy expenditure is obtained by using a smaller percentage of informed nodes. This property has the added benefit of reducing the percentage of total network storage capacity required by each unique agent, decreasing the probability that nodes will need to employ an event table entry replacement protocol. Second, the value of δ'^* for a fixed E_{rcv}/E_{xmt} ratio is approximately proportional to the fourth root of N (i.e., $p_2 \approx 0.265$), indicating that δ'^* increases at a much slower rate than the size of the network. As the E_{rcv}/E_{xmt} ratio increases, b_2 decreases, thus reflecting the increased cost of receiving a transmission.

The value of δ'^* also defines the threshold one-hop neighbor density required to achieve the most energy-efficient search performance. As the average size of a node's neighborhood increases beyond the values indicated in Fig. 6, TSBQ is more efficient than local broadcast (i.e., transmitting the query to all of a node's one-hop neighbors). However, when δ is less than $\delta'^*/(1 - c_1)$, where c_1 is the average proportion of shared neighbors between each QN and PQN, the query should be broadcast to a node's closest neighbors to reduce total energy expenditure. In this case, local flooding is simply a special case of TSBQ in which the computed value of δ'^* is greater than $\delta(1 - c_1)$.

If δ' is decreased below the values in Fig. 6, the expected total energy expenditure increases due to the larger number of query transmissions required to locate an informed node. The unicast query model, in which each query transmission is intended for a single receiver, defines the largest possible reduction in δ', i.e., $\delta' = 1$. The expected total energy expenditure for the unicast rumor routing model, similar to that used in SLR [6], can be computed using (3.11) by substituting $\delta' = 1$. However, analysis of the unicast model indicates much larger values of α are required to achieve the minimum energy expenditure, and the minimum energy expenditure of the unicast model exceeds that of TSBQ. For example, in a 20000-node network with an E_{rcv}/E_{xmt} ratio of 0.7 and $n = 1$, the minimum $E[T]$ of TSBQ consumes 50.2% less energy than the unicast query strategy (338.7 versus 680.0 normalized energy units). Additionally, TSBQ requires only 94 informed nodes per agent to achieve minimum $E[T]$ versus 199 for the unicast protocol, a 52.8% reduction in total network storage capacity consumed per agent. For the 20000-node network, Fig. 8 shows the minimum total energy expended by TSBQ ranges from 45.5% to 75.0% less than trajectory-based unicast search protocols, such as SLR.

Additional analysis of the model reveals the value of α^* increases by a factor of approximately 3.4 for each order of magnitude increase in n (Fig. 9), and δ'^* decreases by a factor of approximately 2.0 for each order of magnitude increase in n (Fig. 10). This result is consistent with intuition: minimum $E[T]$ is achieved by advertising popular data/services to a larger portion of the network, thus per-

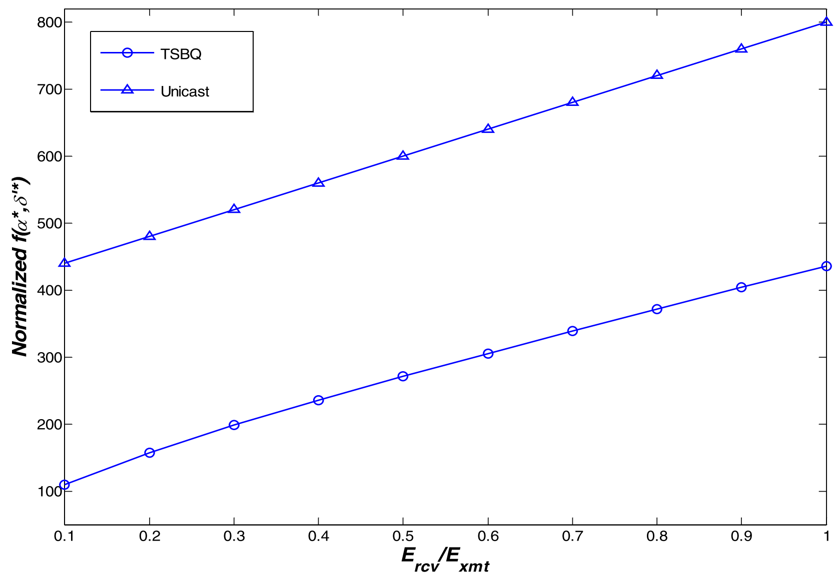


Fig. 8 Minimum $E[T]$ of TSBQ versus unicast search protocols, $N = 20000$, $n = 1$

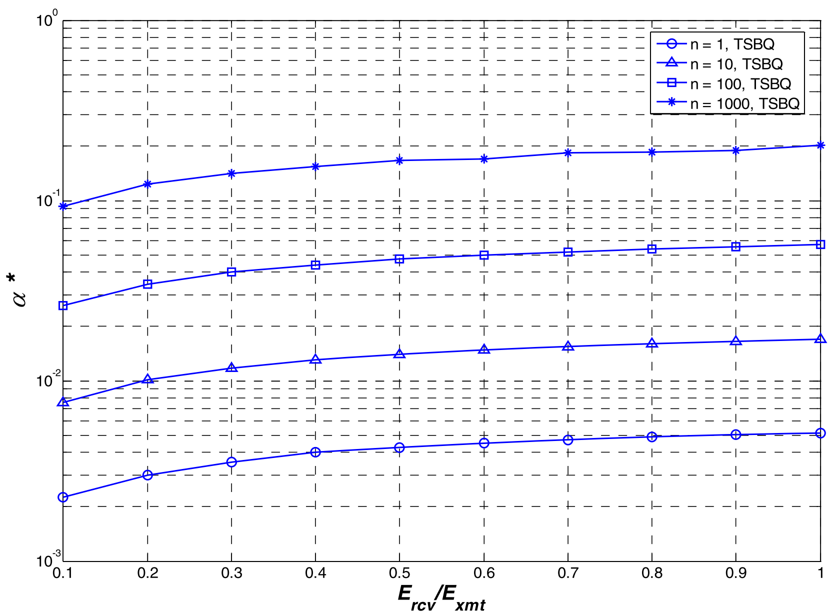


Fig. 9 Effect of n on α^*, TSBQ protocol, $N = 20000$

mitting the energy costs related to advertising to be amortized over a larger number of queries. When a popular item is heavily advertised, we expect to locate the desired information using fewer transmissions. Accordingly, δ' should be decreased to achieve the minimum total energy expenditure when an item is popular and heavily advertised, while δ' should be increased to locate less popular (and, hence, lightly advertised) items.

In contrast to TSBQ, unicast search algorithms require a higher proportion of informed nodes—regardless of the E_{rcv}/E_{xmt} ratio—to achieve minimum $E[T]$. As shown in Fig. 11, the value of α^* for the unicast search protocol is unaffected by the E_{rcv}/E_{xmt} ratio, and this value always exceeds the corresponding α^* value for TSBQ. This is due to the fact that unicast protocols cannot take advantage of efficiencies gained by querying multiple nodes per transmission.

4 Simulation results

We have demonstrated how our mathematical model may be used to minimize the total expected energy expended to locate services and data within a WSN. However, as noted in Sect. 3.2.2, the analytic model makes two simplifying assumptions. First, we assume informed nodes are spatially uniformly distributed throughout the network. Second, the analytic model does not explicitly account for the probability of a query encountering a network boundary prior to

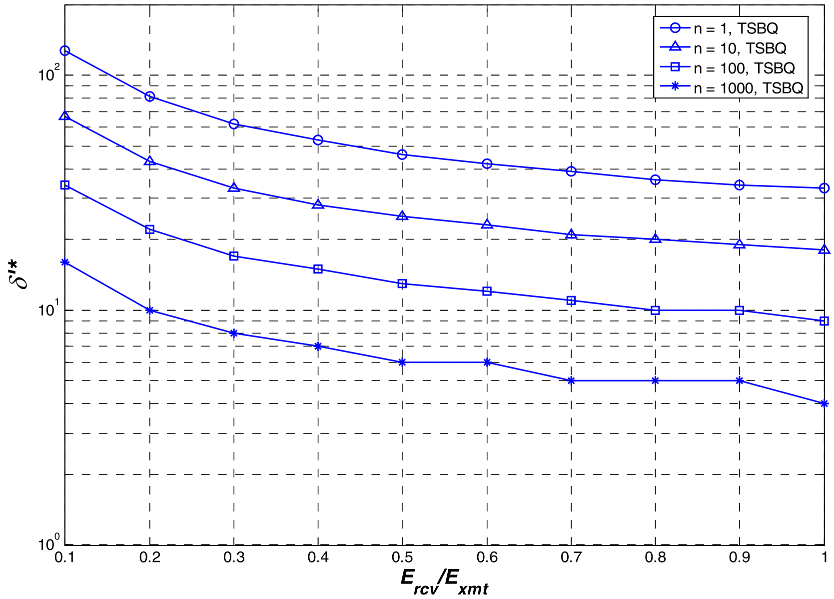


Fig. 10 Effect of n on δ'^*, TSBQ protocol, $N = 20000$

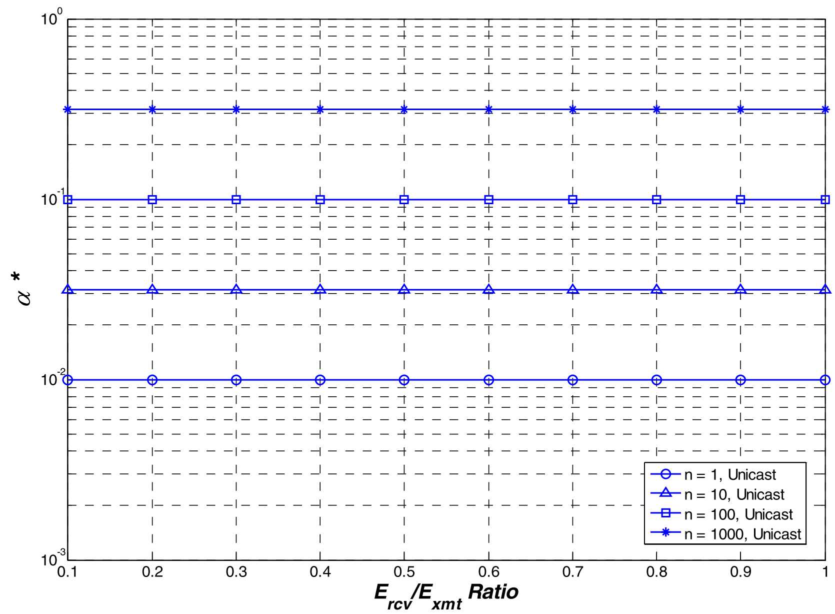


Fig. 11 Effect of n on α^*, unicast search, $N = 20000$

locating an informed node. To examine the significance of these assumptions on our analytic model, we now compare the predicted performance of TSBQ to the results of simulation.

Section 4.1 explains the construction of our network simulator. Section 4.2 examines the impact of network boundaries on the predictive value of our mathematical model, and Sect. 4.3 assesses the effects of trajectory-based forwarding—and the resulting non-uniform distribution of informed nodes—on performance. To improve performance, a simple feedback mechanism is proposed that imposes negligible additional energy cost. Section 4.4 evaluates the predicted and observed variance of energy expenditure per query. Finally, based on our simulation results, Sect. 4.5 proposes an improved mathematical model that incorporates network boundaries.

4.1 Simulation construction

To accommodate the large, dense networks of nodes needed to evaluate the performance of the TSBQ protocol, we develop a network simulation using MATLAB. Since our analytic model assumes a reliable channel, no collisions, and retransmissions managed by the MAC layer (although these effects are indirectly included in the analytic model via the E_{xmt} and E_{rcv} parameters), a MATLAB-based simula-

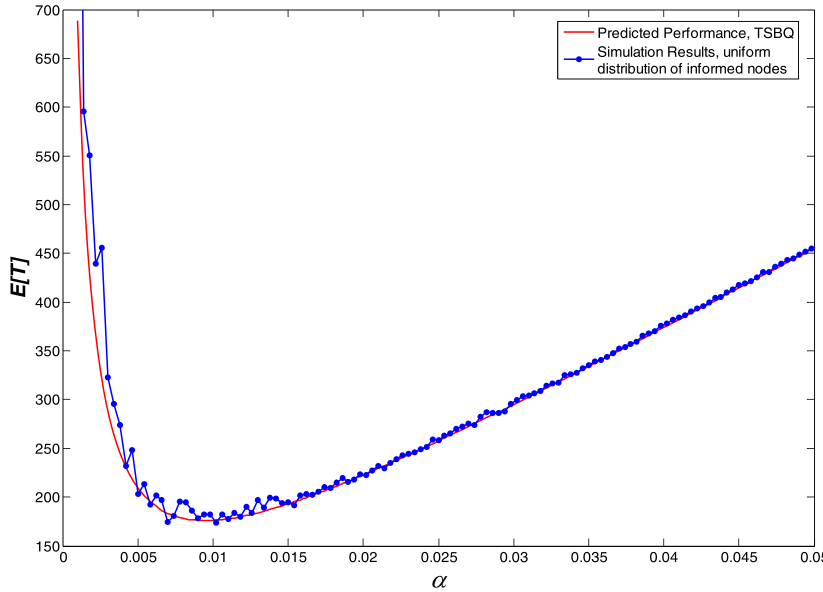


Fig. 12 TSBQ performance, 5000-node network, $\delta' = 27$, $E_{rcv}/E_{xmt} = 0.7$

tion was well-suited for our purposes. Thus, we were able to obtain in a reasonable time 1000 replicates per set of parameters—and ensure the stability of the simulation on a standard desktop PC.

The simulator generates networks of N randomly-placed nodes within the confines of a user-defined square deployment region. To simplify the process of determining the set of neighbors of each node, we assumed a circular radio propagation model and specified the maximum transmission range that results in the minimum acceptable E_b/N_o for each node. Although our transmission model is somewhat unrealistic for indoor environments, it has been found to be accurate for modeling outdoor WSNs [10]. Regardless, TSBQ does not require an isotropic transmission range for proper operation.

The simulation follows the steps of the TSBQ protocol outlined in Sect. 3. First, randomly-selected witness nodes forward an agent to $(\alpha N - 1)$ unique nodes. Once the agents have informed the network, randomly-selected uninformed nodes generate queries. Prior to each query transmission, the transmitting node selects a PQN and also randomly chooses δ' of its closest one-hop neighbors as receiving nodes from among those nodes closer to the current QN than either the PQN or the previous QN. Although our node transmission model results in a well-defined region for choosing RNs (Fig. 2), irregularly-shaped one-hop neighborhoods can be accommodated by permitting designated RNs to turn off their receivers if they determine they have already received a copy of a particular query. Once an informed node is found, the response is returned to the OQN. The mean total energy expended to inform the network, answer each query, and return the response is reported at the completion of 1000 independent trials for each (α, δ') pair. Our simulations consisted of testing 5000-, 10000-, and 20000-node networks using the parameters summarized in Table 1.

Table 1 Simulation model parameters

Network size (N)	Deployment area	Effective node transmission range	Average one-hop neighborhood size (δ)
5000 nodes	30000 m^2	11 m	63
10000 nodes	59395 m^2	11 m	64
20000 nodes	97470 m^2	11 m	78

The average run-time for each simulation varies based on several user-defined parameters, including the number of nodes in the network and the number of replications of each experiment. However, using a 3.2 GHz Pentium IV computer and 1000 replicates per data point, the results presented in the next subsection required approximately 6 hours for the 5000-node network, 17 hours for the 10000-node network, and 56 hours for the 20000-node network.

4.2 Effect of network boundaries on performance

The mathematical model of the expected energy requirement assumes a uniform distribution of informed nodes. Therefore, to study the effect of network boundaries on the performance of our protocol, the simulation was permitted to randomly choose αN informed nodes, thus permitting an assessment of the performance of TSBQ free of the effects of the agent routing method. (The impact of trajectory routing on system performance is evaluated in Sect. 4.3.)

The results of these simulations for 5000-, 10000-, and 20000-node networks are shown in Figs. 12, 13, and 14,

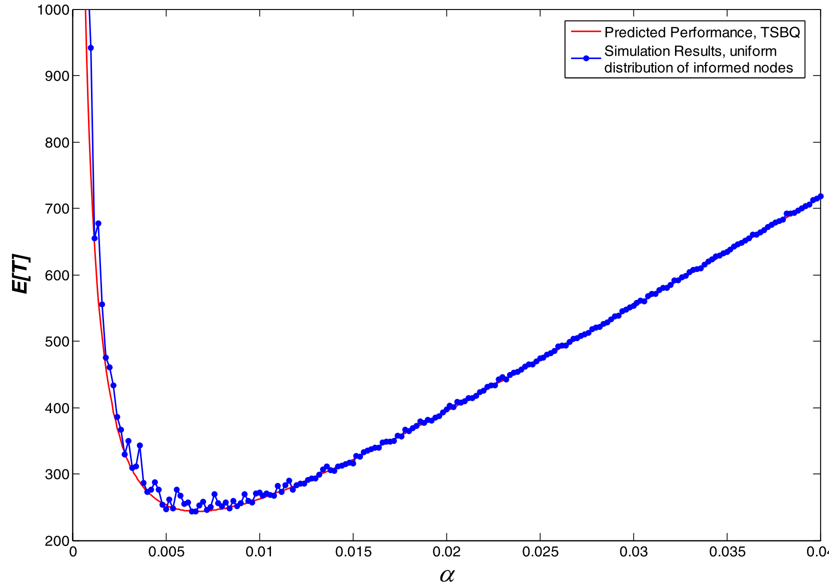


Fig. 13 TSBQ performance, 10000-node network, $\delta' = 32$, $E_{rcv}/E_{xmt} = 0.7$

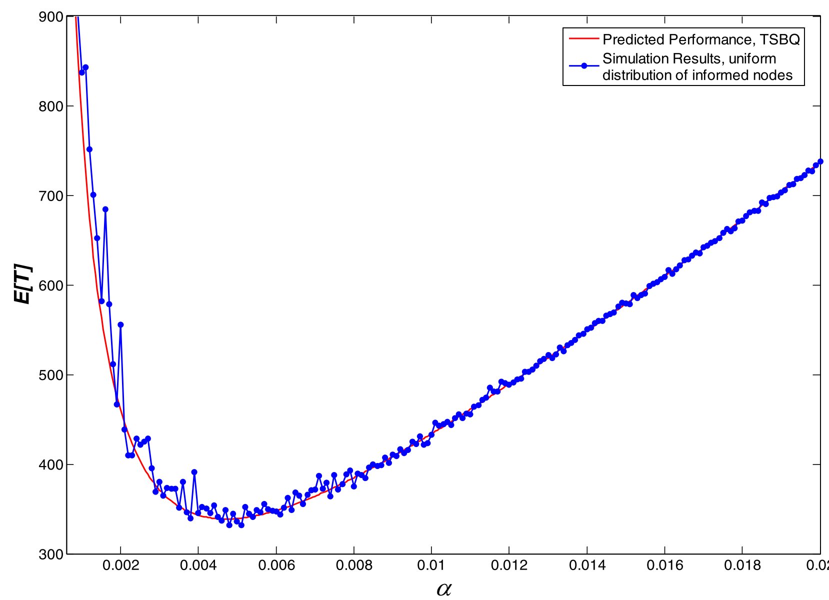


Fig. 14 TSBQ performance, 20000-node network, $\delta' = 39$, $E_{rcv}/E_{xmt} = 0.7$

respectively. Each data point represents the average performance of 1000 independent simulation runs. With the exception of the smallest values of α (e.g., $\alpha < 0.004$ for the 5000-node case), the value of $E[T]$ predicted by (3.11) was within the 95% confidence interval of the simulation results. The observed results at lower values of α differ from the mathematical model due to a large number of queries dropped by the network at a boundary prior to discovering an informed node. When this event occurred in our simulations, the OQN was forced to reissue the query along another randomly-chosen trajectory after an appropriate timeout period. Since we did not limit the OQN's choice of trajectories for reissued queries in the simulation model, a node may receive the same query more than once if subsequent trajectories are similar to the original. We note that TSBQ is designed to prevent nodes from receiving transmissions of the *same* query on subsequent hops. It does not attempt to prevent nodes from being queried more than once by *reissued* queries. However, further energy savings can be obtained by allowing nodes to turn off their receivers once they determine a given query has already been received.

Based on these results, we conclude that the mathematical model is useful for predicting the performance of the network if the actual proportion of informed nodes is not significantly smaller than α^*. However, the predictive capability of the model can be improved at small values of α by extending (3.11) to include parameters associated with the network deployment area and the transmission range of the

nodes. Section 4.5 explains how to extend the mathematical model in this manner.

4.3 Effect of trajectory-based forwarding of agents

Although our mathematical model assumes a spatially uniform distribution of informed nodes, such a distribution of informed nodes is difficult to achieve in real-world networks due to the limited transmission range of nodes. A uniform distribution of informed nodes might be attained by artificially partitioning the network into equal-size zones—such as those used in Zonal Rumor Routing [1]—or by guaranteeing at least k-hop distance between identical event table entries using a method such as k-DID [2], but such schemes require additional energy expenditure and increase complexity. Also, algorithms such as k-DID have been found to scale poorly in dense networks [2]. Instead, we propose to route agents along randomly-chosen straight-line trajectories and use MFR to choose intermediate receivers to achieve maximum initial spatial dispersion of informed nodes in the fewest possible transmissions. As a consequence, we expect mean per-query energy expenditure to differ from that predicted by the mathematical model, especially at lower values of α, due to a spatially non-uniform distribution of informed nodes and queries encountering a network boundary prior to locating an informed node.

To examine the effects of straight-line forwarding of agents on overall energy expenditure, additional simulation experiments were conducted using the parameters in Table 1. The results of these simulations are shown in Figs. 15, 16, and 17. Each data point represents the average performance observed over 1000 independent simulation runs.

As predicted, informing nodes via trajectory-based forwarding results in differences between the predicted and observed mean per-query energy expenditures; however, the general trend of the results follows that predicted by (3.11) at higher values of α. For this reason, we advocate the use of a feedback-driven caching mechanism to increase the number of informed nodes at little or no energy cost to the network. The purpose of this mechanism is to decrease the energy expended by the network to answer future queries; it is also useful if the magnitude of n is unknown during the network design phase.

This feedback-driven caching mechanism is described as follows: once a QN locates an informed node, the actual total number of query transmissions required, $x_{\alpha,\delta'}$, is compared to the number of query transmissions expected, $E[X_{\alpha,\delta'}]$. Assuming the OQN becomes an informed node upon receiving the response, a value ρ, $0 \le \rho \le 1$, is computed by

$$\rho = \max\left\{\frac{x_{\alpha,\delta'} - 2E[X_{\alpha,\delta'}]}{x_{\alpha,\delta'} \cdot E[X_{\alpha,\delta'}]}, 0\right\}. \tag{4.1}$$

Intermediate nodes at each hop in the response's path add the information contained in the response to their own event tables with probability ρ. Although not shown here, our experiments indicate this feedback mechanism provides a significant decrease in total energy expenditure for subsequent queries at the expense of total available network storage capacity. Alternatively, nodes recognizing a higher-than-expected number of queries for a particular agent might also forward the high-demand agent autonomously to inform a larger portion of the network, thereby increasing the probability that additional nodes are capable of answering a query. Additional energy savings may also be realized by aggregating updates.

4.4 Performance variance

The mathematical model and our simulation results indicate the variance in the total energy consumed to generate a response can be large, especially at smaller values of α and δ'. Although we found no mention of a variance analysis of total energy expenditure in the literature, our results can be generalized to any rumor routing-based search algorithm. However, as shown in Fig. 18, the variance of total energy expended (and, hence, the number of transmissions and/or latency required to answer a query) is inversely proportional to α. Therefore, if an application requires a query to be answered within a specific number of transmissions (or, alternatively, specifies a maximum latency) with a given probability, the requirement can be met by adjusting α appropriately. The cost of increasing α, however, is an increase in mean per-query energy consumption and a decrease in the total effective storage capacity of the network. The predicted variance based on the choice of α can be determined by evaluation of the following:

$$\begin{aligned}\mathrm{Var}[X_{\alpha,\delta'}] &= \sum_{j=1}^{k} j^2 \cdot \Pr\{X_{\alpha,\delta'} = j\} \\ &\quad - \left[\sum_{j=1}^{k} j \cdot \Pr\{X_{\alpha,\delta'} = j\}\right]^2.\end{aligned} \tag{4.2}$$

In Fig. 18, the observed variance of T in our simulations is generally higher than that predicted by (4.2) at lower α because a query is dropped if it attempts to travel beyond the defined network boundaries. When a response fails to arrive after the expiration of a timeout period, the OQN may reissue the query along new randomly-chosen trajectories until a response is received. (This is the approach we take in our simulations.) However, if a node chooses random trajectories for reissued queries that result in similar paths through the network, redundant querying of nodes can result. Thus, it may be prudent to limit a node's range of available trajectories in the event that it must reissue a query. Additionally, the predictive value of the model could be improved by incorporating the probability of a query encountering a

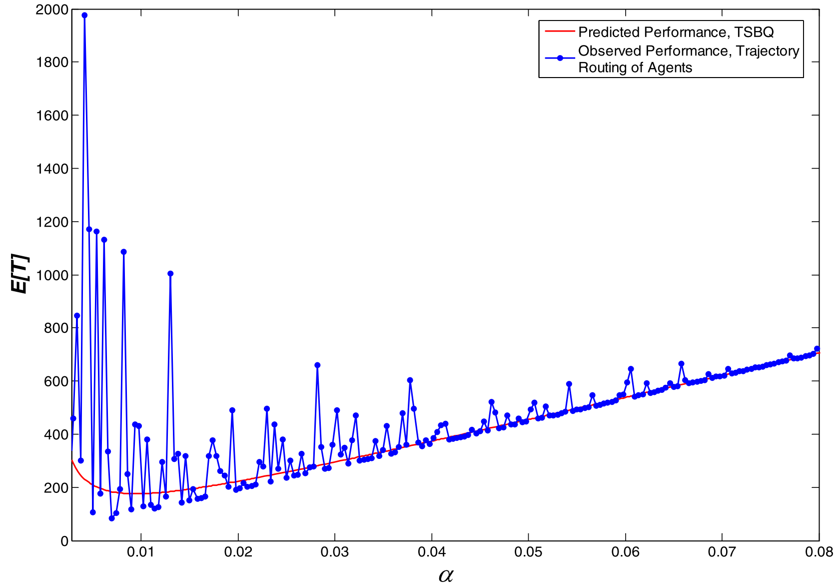


Fig. 15 TSBQ with agent trajectory routing, 5000-node network, $\delta' = 27$, $E_{rcv}/E_{xmt} = 0.7$

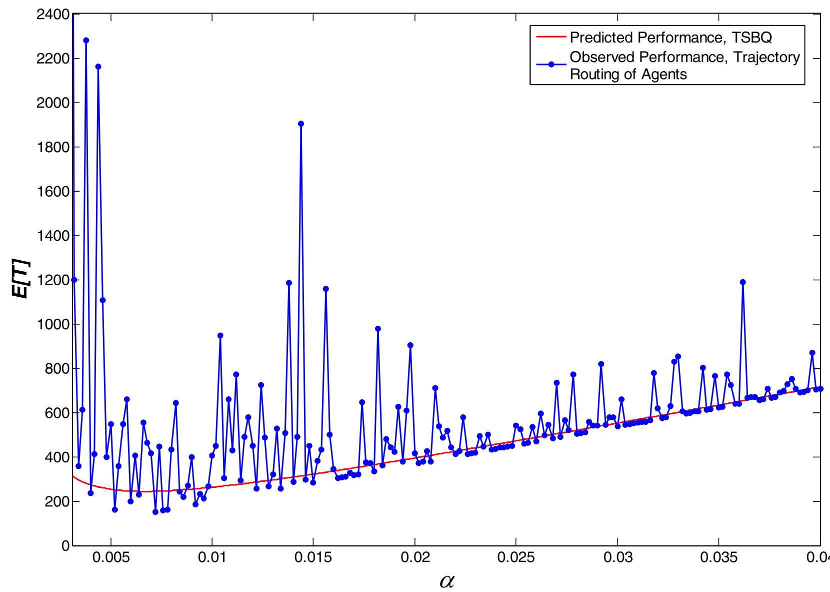


Fig. 16 TSBQ with agent trajectory routing, 10000-node network, $\delta' = 32$, $E_{rcv}/E_{xmt} = 0.7$

network boundary. We discuss this improvement in the next subsection.

4.5 Network boundaries and the analytic model

The mathematical model (3.11) can be improved by accounting for the effect of a query encountering a network boundary prior to locating an informed node. This requires determining the mean hop-distance between a randomly-chosen node and a random point located on the network boundary. If d is the straight-line distance between a randomly-chosen node and a random point on the network boundary, the expected number of hops, β, before a query encounters a boundary is

$$\beta = \left\lceil \frac{d}{R'} \right\rceil \leq k, \tag{4.3}$$

where R' is the mean distance between transmitter-receiver pairs. Assuming a network of sufficient density, R' is approximately equal to the node transmission range R when using MFR routing. The value of d can be determined mathematically or via Monte Carlo experiments. For example, in a square $w \times w$ deployment region such as those used in our simulations, d is approximately $0.65w$. A query that encounters a boundary is expected to have checked $\beta \cdot \delta'$ nodes unsuccessfully. Therefore, the probability of an OQN's original query encountering a network boundary prior to locating an informed node is

Fig. 17 TSBQ with agent trajectory routing, 20000-node network, $\delta' = 39$, $E_{rcv}/E_{xmt} = 0.7$

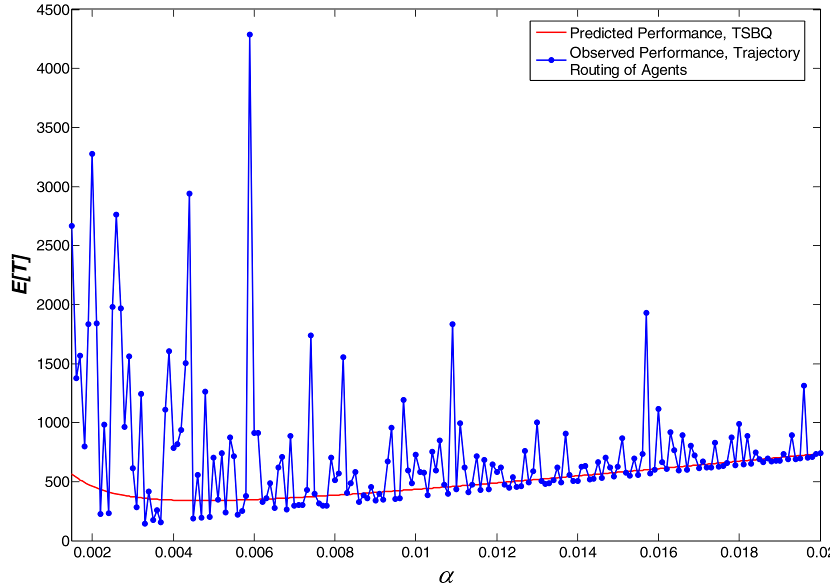
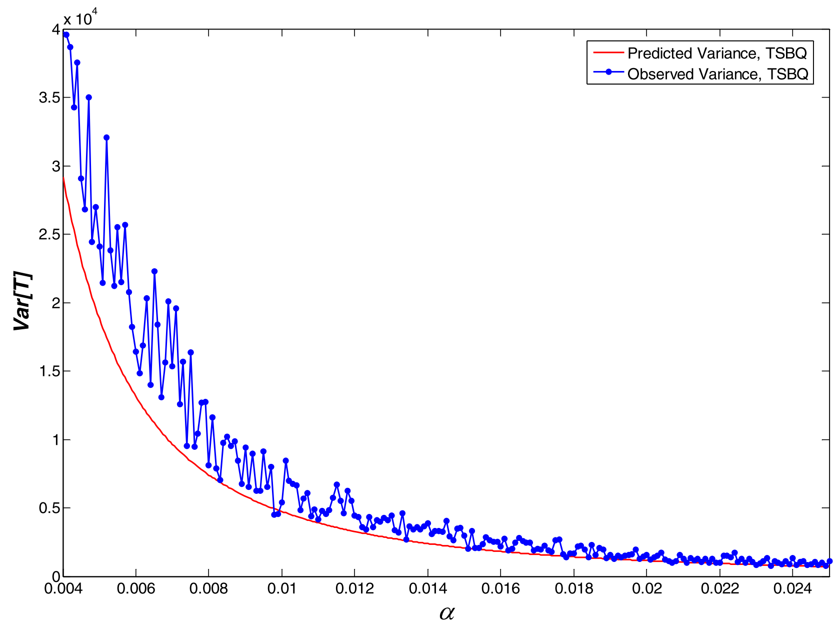


Fig. 18 Predicted vs. observed variance of T, $N = 20000$, $n = 1$, $\delta' = 39$



$$\Pr\{X_{\alpha,\delta'} > \beta\} = \left(1 - \frac{\alpha N}{N-1}\right)^{\beta\cdot\delta'}. \tag{4.4}$$

If the OQN is permitted to reissue failed queries using an unrestricted range of trajectories, the expected number of query attempts, n', to locate an informed node is

$$n' = \left(1 - \left(1 - \frac{\alpha N}{N-1}\right)^{\beta\cdot\delta'}\right)^{-1}. \tag{4.5}$$

Because the OQN's choice of trajectories is not restricted in these experiments, there is a non-zero probability of overlap in the regions of subsequent query transmissions. Therefore, we introduce a term, ζ, to account for the energy expended due to nodes being polled more than once in the event a query is reissued. The value of ζ is a function of both the density and transmission range of the nodes, and $\zeta \geq 1$. Using a least mean squares analysis, the value of ζ for our 20000-node network simulations is approximately 1.438, indicating 43.8% of the nodes polled by all reissued queries received the query transmission more than once. Fortunately, the additional energy expenditure due to repeated polling of nodes is only significant at small values of α. At higher $\alpha, n' \approx 1$; hence ζ has little effect. For example, using the value of α^* shown in Fig. 5 for the 20000-node network, $n' \approx 1.0314$; thus, only 3% of original queries fail to locate an informed node. The revised model for the expected total

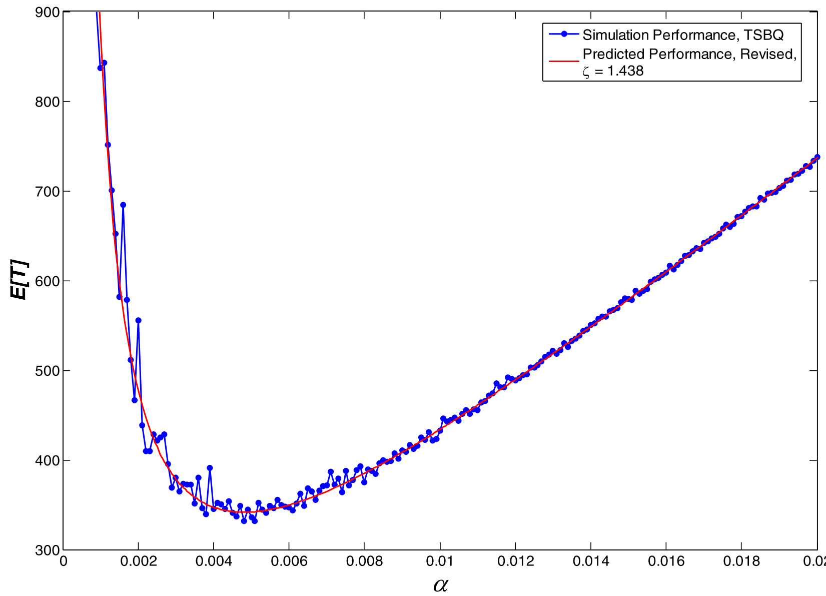


Fig. 19 Revised TSBQ performance, 20000-node network, $n = 1$, $\delta' = 39$

energy expenditure is

$$\begin{aligned} E[T] &= (\alpha N - 1 + n)(E_{xmt} + E_{rcv}) \\ &\quad + (\zeta \cdot n \cdot (n' - 1) \cdot \beta) E_{xmt} \\ &\quad + (\zeta \cdot n \cdot (n' - 1) \cdot \beta \cdot \delta') E_{rcv} \\ &\quad + [2n E_{xmt} + (n \cdot (\delta' + 1)) E_{rcv}] E[X^{\beta}_{\alpha,\delta'}], \end{aligned} \tag{4.6}$$

where $X^{\beta}_{\alpha,\delta'}$ is the expected number of hops required to locate an informed node when network boundaries limit the maximum distance each query may traverse, and

$$E[X^{\beta}_{\alpha,\delta'}] = \sum_{j=1}^{\beta} j \cdot \Pr\{X_{\alpha,\delta'} = j\}.$$

As seen in Fig. 19, (4.6) provides a better prediction of the total energy expended by the network at small α than (3.11). However, (3.11) still provides an accurate means to estimate the values of α^* and δ'^* that result in the least total energy expended without the need to determine ζ.

5 Conclusions and future work

We have developed a mathematical model of a new search protocol, TSBQ, which is used to minimize the total energy expended to advertise services/data and respond to queries in large-scale, high-density WSNs. This protocol is, to the best of our knowledge, the first search protocol to take advantage of the energy efficiency of broadcast transmissions. We developed a mathematical model that predicts the expected total energy expenditure of TSBQ and optimized the model's parameters for minimum energy expenditure. This model enables the network designer to consider the effects of node density, memory capacity, data/service popularity, and latency on the total energy expended to answer a query. Finally, we analyzed the performance variance of TSBQ and provided a feedback-driven caching mechanism that improves search performance at negligible additional energy cost to the network.

The mathematical model of total energy expenditure can be extended to encompass more general search protocols and network application requirements. For example, if a node needs frequent access to a particular service, the most energy efficient strategy is to locate the service in close proximity to the node. The model can be modified accordingly, thereby increasing the probability of locating the service at a nearby node. Additionally, if improved agent dissemination algorithms are developed (i.e., methods that result in a more uniform initial distribution of informed nodes), these algorithms can be incorporated into our model. Finally, the mathematical model can be easily modified to evaluate the optimum transmission range for networks of nodes that have the capability to vary transmission power.

Mobile and three-dimensional networks will become more prevalent and grow in importance as wireless sensor technology advances. However, to date there is little discussion of search algorithm performance in three-dimensional wireless sensor networks. We plan to study the performance of various algorithms, including rumor routing, in such deployment spaces. Furthermore, we see merit in extending our study of search algorithm performance to mobile environments. Finally, we are interested in the effects of finite agent/query lifetimes on optimal replication levels.

Acknowledgements The authors wish to thank the anonymous referees for their helpful comments and suggestions during the development of this paper.

References

1. Banka, T., Tandon, G., & Jayasumana, A. (2005). Zonal rumor routing for wireless sensor networks. In *Proceedings of the international conference on information technology: coding and computing* (Vol. 2, pp. 562–567).
2. Bellavista, P., Corradi, A., & Magistretti, E. (2005). Comparing and evaluating lightweight solutions for replica dissemination and retrieval in dense MANETs. In *Proceedings of the 10th IEEE symposium on computers and communications* (pp. 43–50).
3. Bisnik, N., & Abouzeid, A. (2005). Modeling and analysis of random walk search algorithms in P2P networks. In *Proceedings of the second international workshop on hot topics in peer-to-peer systems* (pp. 95–103).
4. Braginsky, D., & Estrin, D. (2002). Rumor routing algorithm for sensor networks. In *Proceedings of the 1st ACM international workshop on wireless sensor networks and applications* (pp. 22–31).
5. Chen, B., Jamieson, K., Balakrishnan, H., & Morris, R. (2002). Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. *Wireless Networks*, *8*(5), 481–494.
6. Chou, C. F., Su, J. J., & Chen, C. Y. (2005). Straight line routing for wireless sensor networks. In *Proceedings of the 10th IEEE symposium on computers and communications* (pp. 110–115).
7. Cohen, E., & Shenker, S. (2002). Replication strategies in unstructured peer-to-peer networks. In *Proceedings of the 2002 conference on applications, technologies, architectures, and protocols for computer communications* (pp. 177–190).
8. Gaeta, R., Balbo, G., Bruell, S., Gribaudo, M., & Sereno, M. (2005). A simple analytical framework to analyze search strategies in large-scale peer-to-peer networks. *Performance Evaluation*, *62*(1–4), 1–16.
9. Gkantsidis, C., Mihail, M., & Saberi, A. (2005). Hybrid search schemes for unstructured peer-to-peer networks. In *Proceedings of the 24th annual joint conference of the IEEE computer and communications societies* (Vol. 3, pp. 1526–1537).
10. Heidemann, J., Bulusu, N., Elson, J., Intanagonwiwat, C., Lan, K., & Xu, Y. et al. (2001). Effects of detail in wireless network simulation. In *Proceedings of the SCS multiconference on distributed simulation* (pp. 3–11).
11. Karp, B., & Kung, H. T. (2000). GPSR: Greedy perimeter stateless routing for wireless networks. In *Proceedings of the 6th annual international conference on mobile computing and networking* (pp. 243–254).
12. Liu, D., Hu, X., & Jia, X. (2006). Energy efficient protocols for information dissemination in wireless sensor networks. In *Lecture notes in computer science* (Vol. 3842, pp. 176–185).
13. Lu, G., Krishnamachari, B., & Raghavendra, C. (2004). An adaptive energy-efficient and low-latency MAC for data gathering in sensor networks. In *Proceedings of the 18th international parallel and distributed processing symposium* (p. 224).
14. Malkhi, D., Naor, M., & Ratajczak, D. (2002). Viceroy: A scalable and dynamic emulation of the butterfly. In *Proceedings of the 21st annual symposium on principles of distributed computing* (pp. 183–192).
15. Manku, G. S., Naor, M., & Wieder, U. (2004). Know thy neighbor's neighbor: The power of lookahead in randomized P2P networks. In *Proceedings of the 36th annual ACM symposium on theory of computing* (pp. 54–63).
16. Melamed, R., Keidar, I., & Barel, Y. (2005). Octopus: A fault-tolerant and efficient ad-hoc routing protocol. In *Proceedings of the 24th IEEE symposium on reliable distributed systems* (pp. 39–49).
17. Nath, B., & Niculescu, D. (2003). Routing on a curve. *ACM SIGCOMM Computer Communication Review*, *33*(1), 155–160.
18. Rajendran, V., Obraczka, K., & Garcia-Luna-Aceves, J. J. (2006). Energy-efficient, collision-free medium access control for wireless sensor networks. *Wireless Networks*, *12*(1), 63–78.
19. Ratnasamy, S., Francis, P., Handley, M., Karp, R., & Schenker, S. (2001). *A scalable content-addressable network*. New York: ACM Press.
20. Ratnasamy, S., Karp, B., Yin, L., Yu, F., Estrin, D., & Govindan, R. et al. (2002). GHT: A geographic hash table for data-centric storage. In *Proceedings of the 1st ACM international workshop on wireless sensor networks and applications* (pp. 78–87).
21. Ratnasamy, S., Karp, B., Shenker, S., Estrin, D., Govindan, R., & Yin, L. et al. (2003). Data-centric storage in sensornets with GHT, A geographic hash table. *Mobile Networks and Applications*, *8*(4), 427–442.
22. Rowstron, A., & Druschel, P. (2001). Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In *IFIP/ACM international conference on distributed systems platforms (middleware)* (pp. 329–350).
23. Shenker, S., Ratnasamy, S., Karp, B., Govindan, R., & Estrin, D. (2003). Data-centric storage in sensornets. *Computer Communication Review*, *33*(1), 137–142.
24. Stoica, I., Morris, R., Karger, D., Kaashoek, M. F., & Balakrishnan, H. (2001). Chord: A scalable peer-to-peer lookup service for internet applications. *Proceedings of the 2001 SIGCOMM conference*, *31*(4), 149–160.
25. Stojmenovic, I. (1999). *A scalable quorum based location update scheme for routing in ad hoc wireless networks* (TR 99-09). SITE, University of Ottawa.
26. Stojmenovic, I., & Lin, X. (2001). Loop-free hybrid single-path/flooding routing algorithms with guaranteed delivery for wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, *12*(10), 1023–1032.
27. Tchakarov, J. B., & Vaidya, N. H. (2004). Efficient content location in wireless ad hoc networks. In *Proceedings of the 2004 IEEE international conference on mobile data management* (pp. 74–85).
28. van Dam, T., & Langendoen, K. (2003). An adaptive energy-efficient MAC protocol for wireless sensor networks. In *Proceedings of the 1st international conference on embedded networked sensor systems* (pp. 171–180).
29. Xu, Y., Heidemann, J., & Estrin, D. (2001). Geography-informed energy conservation for ad hoc routing. In *Proceedings of the 7th annual international conference on mobile computing and networking* (pp. 70–84).
30. Xue, F., & Kumar, P. R. (2006). On the θ-coverage and connectivity of large random networks. In *IEEE/ACM Transactions on Networking (TON)*, (Vol. 14, pp. 2289–2299).
31. Ye, W., Heidemann, J., & Estrin, D. (2002). An energy-efficient MAC protocol for wireless sensor networks. In *Proceedings of the 21st annual joint conference of the ieee computer and communications societies* (Vol. 3, pp. 1567–1576).
32. Zhang, H., & Hou, J. (2004). On deriving the upper bound of α-lifetime for large sensor networks. In *Proceedings of the 5th ACM international symposium on mobile ad hoc networking and computing* (pp. 121–132).
33. Zhao, B. Y., Kubiatowicz, J., & Joseph, A. D. (2001). *Tapestry: An infrastructure for fault-tolerant wide-area location and routing* (Tech. Rep. UCB/CSD-01-1141). Computer Science Division, University of California, Berkley.

Christopher R. Mann is an Air Force officer and Ph.D. candidate at the Air Force Institute of Technology. He earned the B.S.E.E. from the United States Air Force Academy in 1991, the M.S.E.E. from the Georgia Institute of Technology in 1992, and the M.A. in Organizational Management from the George Washington University in 1998. He is also an Air Force pilot with nearly 2,500 hours of flight time in the KC-135, C-17, and C-12 aircraft. He is a member of Eta Kappa Nu and Tau Beta Pi. His research interests include wireless sensor networks, digital communications, and queuing theory.

Rusty O. Baldwin is an Associate Professor of Computer Engineering in the Department of Electrical and Computer Engineering, Air Force Institute of Technology, Wright-Patterson AFB OH. He received a B.S. in Electrical Engineering (cum laude) from New Mexico State University in 1987, an M.S. in Computer Engineering from the Air Force Institute of Technology in 1992, and a Ph.D. in Electrical Engineering from Virginia Polytechnic Institute and State University in 1999. He served 23 years in the United States Air Force. He is a registered Professional Engineer in Ohio, a member of Eta Kappa Nu, and a Senior Member of IEEE. His research interests include computer communication networks, embedded and wireless networking, computer security, information assurance, and reconfigurable computing systems.

Jeffrey P. Kharoufeh is an Associate Professor of Operations Research at Northeastern University. He holds B.S. and M.S. degrees in Industrial and Systems Engineering from Ohio University and a Ph.D. in Industrial Engineering and Operations Research from Penn State University. Dr. Kharoufeh's field of specialty is applied probability and stochastic processes with applications in queuing systems, reliability theory, and maintenance optimization. He serves as an Associate Editor for the IEEE Transactions on Reliability and Operations Research Letters.

Barry E. Mullins is an Assistant Professor of Computer Engineering in the Department of Electrical and Computer Engineering, Air Force Institute of Technology, Wright-Patterson AFB OH. He received a B.S. in Computer Engineering (cum laude) from the University of Evansville in 1983, an M.S. in Computer Engineering from the Air Force Institute of Technology in 1987, and a Ph.D. in Electrical Engineering from Virginia Polytechnic Institute and State University in 1997. He served 21 years in the Air Force teaching at the U.S. Air Force Academy for seven of those years. He is a registered Professional Engineer in Colorado and a member of Eta Kappa Nu, Tau Beta Pi, IEEE (senior member), and ASEE. Dr. Mullins has received the U.S. Air Force Academy's Outstanding Academy Educator award as well as the Brig. Gen. R. E. Thomas award for outstanding contribution to cadet education twice. His research interests include computer communication networks, embedded (sensor) and wireless networking, cyber operations, and reconfigurable computing systems.